Software Reviews Since Acquisition Reform – The Artifact Perspective

Dr. Peter Hantos

Senior Engineering Specialist

Software Acquisition and Process Office

Acquisition of Software Intensive Systems Conference 2004

THE AEROSPACE CORPORATION

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JAN 2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Software reviews Since Acquisition Reform - The Artifact Perspective	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The Aerospace Corporation,PO Box 92957,Los Angeles,CA,90009-2957**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
2004 Conference on the Acquisition of Software-Intensive Systems, 26-28 Jan, 2004.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**17**	

Acknowledgements

- This work would not have been possible without assistance from the following:
  - Reviewers
    - Richard J. Adams, Software Acquisition and Process Office
    - Suellen Eslinger, Software Acquisition and Process Office
    - Karen L.Owens, Software Acquisition and Process Office
    - Mary A. Rich, Principal Director, Software Engineering Subdivision
  - Sponsor
    - Michael Zambrana, USAF Space and Missile Systems Center, Directorate of Systems Engineering
  - Funding source
    - Mission-Oriented Investigation and Experimentation (MOIE) Research Program (Software Acquisition Task)

Background of Problem

- **Pre-1994:**
  - MIL-STD-1521B (Technical Reviews)
    - Formal milestone reviews
    - Date of last version is June 4, 1985 (!)
  - DoD-STD-2167A (Defense System Software Development)
    - Single pass Waterfall Life Cycle Model bias
- **1994:**
  - MIL-STD-498 (Software Development & Documentation)
    - Although all other MIL standards are cancelled by the DoD, **MIL-STD-498** was approved as an interim standard for 2 years
    - Eliminated any Waterfall bias
    - Joint reviews: Schedule and content proposed by contractor

The Problems with Reviews

- **Now:**
  - No official development or review standards of record
  - Neither the government nor the contractor has clear concept of what reviews should contain and when they should occur
  - Interpretation of major contractual technical reviews (e.g., System PDR, System CDR) is left to individuals to decide
  - Quality and content of reviews is widely different both within and across programs
  - Quick, last-minute, before-review efforts to revive and customize MIL-STD-1521B proved to be ineffective

Perspectives on Review Issues

- **The Life Cycle Perspective ("When?")***
  - Pre-acquisition reform assumptions:
    - Acquisition and development are exclusively Waterfall
    - Reviews (SSR, PDR, CDR, etc.) are clearly positioned
  - Now:
    - Evolutionary Acquisition
    - Iterative/Incremental and Spiral Development
    - Emerging agile methods
    - Asynchronous, in-process, interim reviews
- **The Artifact Perspective ("What?")**
  - This is the subject of the presentation

**** For more details see my upcoming presentation at the 2004 Software Technology Conference in Salt Lake City, Utah: Hantos, P.,"Software Reviews Since Acquisition Reform – The Life Cycle Perspective"***

Presentation Objectives

- **Identify** modern software development trends within key areas of interest
- **Compare** pre-acquisition reform software development practices with the state-of-the-practice (With minor references to the state-of-the-art…)
- **Highlight** new work products and related, new review artifacts

Key Areas of Interest

Architecture	Unit Designation and Nomenclature
	Architecture Focus
	Software Reuse and COTS
	Frameworks
	Open Systems
	Distributed Systems
Product-Oriented Software Engineering Activities	Analysis/Design
	Programming Languages
	Programming
	Integration
	Test
Engineering Management Process	The Software Concept
	Quantitative Management, use of Software Metrics
	Organizational and Management Models
	Systems Engineering
Integral Software Engineering Activities	Process Maturity and Quality Frameworks
	Quality
	Risk Management
Hardware-Software Technology	Design Paradigms
	Host Processor
	Communications
	Database Management
	Tools
	Documentation
Security	Security

Architecture

	OLD	NEW
Unit Designation and Nomenclature	**CSCI – HWCIs with high granularity:** • Homogeneous, static view of CSCIs assuming unchanging configuration entities: *Design -> Source Code ->* *-> Developmental executables ->* *-> Delivered executables*	**O-O concepts and nomenclature:** Objects – with flexible granularity Packages – for higher order, logical object structure Component diagrams – for components and interfaces Deployment – distribution of components on nodes Source code vs. executable releases
Architecture Focus	**Weak:** • High-level Design = Architecture **Monolithic architectural patterns:** • Mainframe • Process control-type instruments	**Strong:** Multiple,stakeholder views **Variety of architectural patterns:** Client-Server Distributed/networked Application-services
Software Reuse and COTS	**Sporadic, opportunistic reuse** **Limited, low-level libraries only** **No sensitivity to COTS software:** • No impact on life cycle models • No acknowledgement of risks due to COTS volatility	**Systematic, application-domain reuse** **Increased use of system libraries** **High emphasis on using COTS:** Acknowledging life cycle impact High sensitivity to COTS volatility Coexistence with legacy code, "wrappers"

Architecture (Cont.)

	OLD	NEW
Frameworks	**Middleware concept didn't exist**	**Wide use of frameworks:** COM, .NET, CORBA, etc. Distributed objects Domain-specific, reusable assets
Open Systems	**Concept didn't exist** • Most solutions were proprietary • Limited acceptance of best practices	**Rapidly emerging concept** Push for commercial solutions Open to wide array of best practices Emerging, Open UML-based standards
Distributed Systems	**Primitive networks only** • No WWW (World-Wide Web)	**Large, high bandwidth networks** 100 Gbit/sec Rapidly growing WWW Intranets/Extranets Remote network management Remote diagnostics

Product-Oriented SW Engineering Activities

	OLD	NEW
Analysis/Design	**Structured/Hierarchical:** • Functional decomposition of requirements • No or little coding until design completed	**Object-Oriented/UML-based:** Iterative/Incremental Use Case driven Exploratory coding, prototyping Might follow test-driven design process
Programming Languages	**Primarily procedural** **Textual only**	**Dominant Object-Oriented** **Visual languages**
Programming	**Manual only** **Code metrics:** • McCabe complexity	**Visual programming** **Automated code generators:** Executable models Round-trip engineering **Code metrics for O-O programs:** New metrics replace McCabe
Integration	**"Big-bang":** • Single, major event at the end	**Incremental integration:** Multiple, frequent releases
Test	**Manual only**	**Increased automated testing** **Load testing of networks**

Engineering Management Processes

	OLD	NEW
The Software Concept	**"Software is software"** **"One size fits all":** • Scaling assumed to be simple and transparent	**Acknowledges the differences between:** Information Management (IM) Decision-making systems Real-time applications Web-services Etc. **High sensitivity to scaling**
Quantitative Management, use of SW Metrics	**Weak management exploitation of metrics** **Sporadic choice and use of metrics**	**Systematic use of software metrics** **Strong emphasis on moving to statistical software process control:** CMMI® Level 4-5 Six Sigma
Organizational and Management Models	**Functional organization structure** • Matches the hierarchically decomposed product architecture **No sensitivity to multi-contractor environments**	**Organization structured around IPTs (Integrated Product Teams)** **Acknowledges highly diverse and complex contractor structure**
Systems Engineering	**Weak, nominal software representation and awareness:** • Software always the bottleneck • Disjointed hardware-software processes	**Stronger awareness of software processes and dependencies:** Integrated hardware-software planning Concurrent Engineering

Integral Software Engineering Activities

	OLD	NEW
Process Maturity and Quality Frameworks	**Did not exist**	**CMM/CMMI®:** Organizational process capability and maturity concepts Clearly defined process areas with detailed practices **ISO 9000 series of quality standards**
Quality	**Software Quality = QA + QC = Test** **Emphasis on system test** • Equivalent of QC **Focus on product quality only**	**"New" Software Quality Assurance:** Spread over the life cycle Peer Review of all requirements, design, coding, and test artifacts Emphasis is on defect prevention Scope now includes process audit and process improvement coaching
Risk Management	**Hardware-focused:** • Little sensitivity to and awareness of software risks • Assumes hardware-like reliability models for software • Hardware-software risks handled separately	**Integral activity of Spiral Development:** Hardware-software risk mitigation related trade-offs must be done together

Hardware-Software Technology

	OLD	NEW
Design Paradigms	**Single, basic software paradigm**	**New, emerging paradigms:** Object-Oriented Agents Genetic Programming Neural Networks
Host Processor	**Single processor**	**Multi-processor:** Multi-threaded applications Networks of processors (Grid computing)
Communications	**Low bandwidth:** • No significant compression • Limited wireless communication	**High bandwidth:** Sophisticated compression technologies High-speed wireless communication
Database Management	**Mainframe oriented:** • Large • Text only • At most relational schema	**On any Host:** Fully scalable Distributed O-O for any objects (image, voice, video)
Tools	**Basic, independent tools** **No COTS software perspective**	**Rich toolset, integrated with life cycle process** **High tools vendor/product volatility sensitivity**
Documentation	**All documentation static** **Delivered on paper** **Text format primarily ASCII** **No electronic page formatting**	**Dynamic, interactive, searchable** **Executable models for design** **New media (CD, DVD, or on-line server-based)** **New formats (PDLs, XML)**

Security

	OLD	NEW
Security	**No sensitivity to software specifics:** • Satisfied with password level • System security view only	**High Security Consciousness:** Kernel level security Malicious penetration and virus issues Firewalls, honeypots Denial of Service attacks Trusted computer platforms Sophisticated encryption algorithms Digital Rights management

Conclusions

- In-process software technical reviews are replacing rigid milestone reviews
- The understanding of software development trends is essential for determining review artifacts and their expected performance and maturity according to their position in the system life cycle
- Key Areas of Interest:
  - Architecture
  - Product-Oriented Software Engineering Activities
  - Engineering Management Processes
  - Integral Software Engineering Activities
  - New Hardware-Software Technologies
  - Security

Acronyms and Abbreviations

ASCII	American Standard Code for Information Interchange
CD	Compact Disc
CDR	Critical Design Review
CMMI	Capability Maturity Model Integration
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-the-Shelf
CSCI	Computer Software Configuration Item
DVD	Digital Video Disc
HWCI	Hardware Configuration Item
IPT	Integrated Product Team
ISO	International Organization for Standards
.NET	Microsoft's Web-services Framework
O-O	Object-Oriented
PDL	Page Definition Language
PDR	Preliminary Design Review
QA	Quality Assurance
QC	Quality Control
SSR	System Specification Review
UML	Unified Modeling Language
USAF	US Air Force
WWW	World Wide Web
XML	Extensible Markup Language

Contact Information

Peter Hantos
The Aerospace Corporation
P.O. Box 92957-M1/112
Los Angeles, CA 90009-2957
Phone: (310) 336-1802
Email: peter.hantos@aero.org